REMARKS/ARGUMENTS:

Claims 1-4, 7-24, and 27-39 have been amended in light of the rejection of claims 1-39 in light of Donohue.

As noted by the Examiner, Donohue broadly teaches a method of upgrading software of a computer system. Like the present invention, upgrading per Donohue is similar to adding a new component to a multi-component electronic system per the present invention.

Nevertheless, the present invention differs from the upgrading of Donohue in that the present invention contemplates a system where added components (as part of the upgrade) must share resources with other components and find and load necessary shared resources if they are not present. In Donohue the upgraded components require and search for an "updater" which is downloaded but is not shared with other components but rather is unique to the upgraded component. Thus the "updater" searched for by Donohue is not a shared resource within the system as required by independent claims 1, 21, and 30.

Separately, Donohue considers the problem of "pre-requisite" software which apparently can include software shared among components of the system. Pre-requisite software in Donohue, however, does not appear to be installed into the system as required of claims 1 and 21, but simply upgraded if necessary (see generally column 16, lines 6-26.) Accordingly, Donohue does not address the significant problems of compatibility among shard components.

More fundamentally, Donohue makes the assumption (unwarranted in the case of shared resources) that the "pre-requisite" software may be freely upgraded without affecting other components that might not be compatible with the upgrade. Recognition that this assumption of Donohue is incorrect for shared components underlies an object of the present invention (see page 5, lines 26-28). Thus Donohue teaches away from claims 7 and 27 which address this problem unique to shared resources by teaching a sophisticated adding of components without removing other similar components (for example, having an earlier version number).

Further, because Donohue contemplates upgrading rather than adding or removing of shared components, there is no provision in Donohue for deleting shared components per claims 18, 19, and 20, and claims 37, 38, and 39, and in particular, there is no teaching description of how this can be done without disabling other components that might rely on the deleted or shared resources being deleted or suggestion that it can be done.

As noted by the Examiner, Curtis teaches a system for preventing the deletion of shared resources when they are used by a currently installed component, but even together, Donohue and Curtis fail to teach searching for shared resources that are not on the system and loading them onto the system as required by the independent claims. Donohue limits its searching to unique updaters and other components where conflicts (like those inherent with shared resources are not an issue). Curtis doesn't search and download at all, but assumes that all the shared resources needed during an installation are packaged with the installer or on the system. If not, the installer simply fails (col. 12, lines 17-20). A person of ordinary skill in the art reading Donohue and Curtis would correctly assume that the problems of automatically searching for and installing shared resources have not been solved.

In light of these amendments and comments it is believed that claims 1-39 are now in condition for allowance, and allowance is respectfully requested.

Respectfully submitted,

REGINALD W. SPRECHER, *et al.*

By: _____

Keith M. Baxter
Reg. No. 31,233
Attorney for Applicant
Quarles & Brady LLP
411 E. Wisconsin Avenue
Milwaukee WI 53202-4497
(414) 277-5719